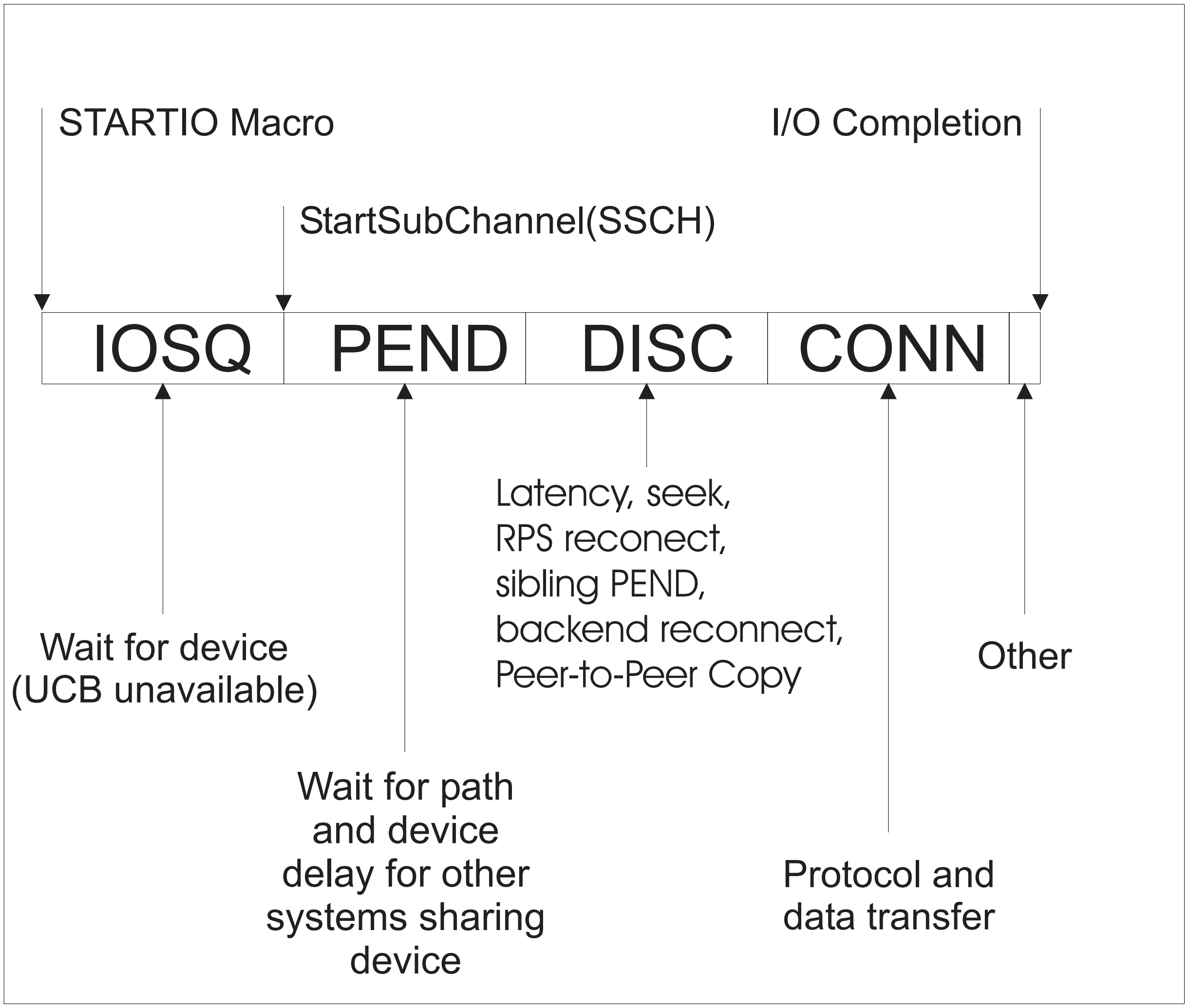
Section 5: DASD Analysis Factors

This section discusses DASD performance analysis considerations. Chapter 1 presents an overview of performance factors from a DASD I/O operation viewpoint. Chapter 2 highlights some of the factors that must be considered when analyzing DASD performance based upon data collected and recorded by SMF or RMF.

Chapter 1: Overview of DASD Performance Considerations

From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. These measures are reported by RMF in SMF Type 74 records. Exhibit 5-1 illustrates these four measures and another potential element of DASD I/O time, titled "Other".



MAJOR COMPONENTS OF DASD I/O OPERATIONS

EXHIBIT 5-1

Chapter 1.1: IOSQ time

IOSQ time is the time from the issuance of a STARTIO macro until the StartSubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the device is busy with *this system*; that is, whether there is an available Unit Control Block (UCB) for the device. If the device is not busy with *this system* (a UCB is available), the SSCH instruction is issued. However, if the device is busy with *this system*, the I/O request is queued. Thus, IOSQ time always means that the device is unable to handle additional requests from *this system*. (The emphasis on "this system" is explained in the below discussion of PEND time.)

This discussion of IOSQ time does not always apply to Parallel Access Volumes (PAVs)[1]. With PAV devices, MVS creates multiple UCBs for each device, depending on how many “alias devices” have been defined. The multiple UCBs allow multiple active concurrent I/Os on a given device when the I/O requests originate from the same system[2]. Using PAVs can dramatically improve I/O performance by nearly eliminating IOSQ.

Some small IOSQ time is often unavoidable. However, large IOSQ time imply a situation that should be examined. Large IOSQ times result from (1) too many I/O operations directed to the device or (2) lengthy device response times (perhaps caused by low percent cache hits or high PEND time). Large IOSQ times usually involve the following situations:

- Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing for the single volume. Alternatively, using the Parallel Access Volume feature available with IBM’s Enterprise Storage Server (ESS) could allow multiple concurrent access to the device.

- Multiple users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the IOSQ problems. Alternatively, using the Parallel Access Volume feature available with IBM’s Enterprise Storage Server (ESS) could allow multiple concurrent access to the device.

- Multiple application systems may be using the volume experiencing high IOSQ times. In this case, perhaps application redesign or scheduling can solve the problem. Alternatively, using the Parallel Access Volume feature available with IBM’s Enterprise Storage Server (ESS) could allow multiple concurrent access to the device.

- A particular application (or system function) may be executing I/O to the device faster than the device can respond. Using application features as Data In Memory, increased buffering, using Local Shared Resources (LSR) or increasing buffer sizes, specifying optimal buffering parameters, and other similar enhancements could allow the applications to considerably reduce the use of I/O activity.

- The overall device response time (PEND, DISC, and CONN) times may be large, such that the device is unable to provide quick response to the I/O requests. This situation will be revealed by large values in the PEND, DISC, or CONN measures.

[1] PAV devices are available with Enterprise Storage Server (ESS). With PAV devices, a “base device” address is defined, and a UCB is associated with this base address. “Alias device” addresses can be defined and UCBs are associated with the alias device addresses.

[2] Multiple Allegiance allows multiple active concurrent I/O operations on a given device when the I/O requests originate from different systems.

With Parallel Access Volumes (PAV) and dynamic alias management in Goal Mode, IOSQ time should be significantly reduced or eliminated. The implications of PAV and dynamic alias management will be discussed in rules related to these features.

Chapter 1.2: PEND time

PEND time is the time from the issuance of the StartSubChannel (SSCH) instruction until the device is selected by the control unit and physical positioning commands (such as seek and set sector) are transferred to the device. With modern fixed block architecture (FBA) devices, the PEND time ends when the physical positioning commands are presented to the *logical volume control block* within the control unit. The PEND time is caused by queuing for the path (wait for channel, wait for director port, wait for control unit, wait for device, or wait for “other” reasons)[3].

The PEND time can be caused by the device being busy from *another system*. In this case, the system issuing the STARTIO macro (*this system*) would have no knowledge that the device was busy with another system. Rather, if a UCB were available for the device, the SSCH would be issued. However, the device could not necessarily be selected (unless multiple allegiance were available), since the device would be busy from another system.

Additionally, PEND time could accumulate even with PAV devices if the access were to an xtent that was busy with another I/O operation from *this system*.

Large PEND times usually involve the following situations:

- **Shared devices**. If the device is shared with another system, PEND time may indicate contention with the other system. Large PEND times in shared-device environments usually involve situations very similar to those described under IOSQ time:

  - Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing at the channel level (reflected as PEND time) for the single volume.

    Alternatively, if IBM’s Enterprise Storage Server (ESS) is available, the Multiple Allegience feature can be used to significantly reduce or eliminate PEND time caused by other systems. Multiple Allegiance allows multiple active concurrent I/O operations on a particular device when the I/O requests originate from different

[3] PEND time is significantly reduced with FICON channels. FICON channels can have multiple I/O operations concurrently active, which reduces the potential PEND time caused by channel busy. There is no port busy time with FICON switches, and control unit time is significantly reduced. This statement regarding PEND time is not necessarily correct if a large number (more than 5) I/O operations are concurrently executing on a FICON channel. Dr. H. Pat Artis and Mr. Robert Ross have presented the results of research indicating that performance degrades significantly when more than 5 I/O operations (Open Exchanges) are concurrently active on a FICON channel (see “Understanding FICON Channel Path Metrics”at www.perfassoc.com).

systems. With Multiple Allegiance, there is complete access with read I/O operations. For write I/O operations, there is concurrent access unless there is a conflicting extent[4]. If there is a conflicting extent, the controller holds the I/O operation in a PEND state for the device.

If some of the data sets are not required to be shared, then the Data Base Administrator has complete flexibility to move these data sets (subject, of course, to the performance implications of the target devices). These data sets could be moved to a non-shared device.

- Multiple applications or users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set may be placed on different volumes. This would solve the PEND problems cause by contending systems. If this option is feasible, the data sets could be placed on non-shared devices, likely resulting in even more performance improvement. Alternatievly, Record-Level Sharing (RLS) might provide a substantial reduction in the exclusive use of data sets.

- Multiple application systems may be using the volume experiencing high PEND times. In this case, perhaps application redesign or scheduling can solve the problem.

- **Non-shared devices**. Large PEND times for devices that are not shared may mean that there are insufficient paths available to the device. Too much I/O may be directed to many devices on the path, control unit, or. The data sets can be redistributed among different logical volumes on different paths, control units, or devices. This will reduce the hardware-level queuing. Alternatively, the entire volume may be moved to a different (less busy) path.

  If redistributing the data sets or moving the logical volume is not feasible, then the device should have more paths. Depending upon the existing configuration, this may involve re-configuring existing channel paths, or acquiring additional hardware.

  Fortunately, SMF Type 78 and Type 74 records contain information that can be used to identify at which level the hardware queuing occurs (that is, whether queuing is for the director port, control unit, or deivce; and CPExpert calculates an estimate of the PEND time caused by channel activity).

- **Devices attached to cached controllers**. Large PEND times for devices attached to cached controllers may imply a high percent of read miss operations, or non-volatile storage (NVS) writes for IBM-3990-3 devices.

[4] A conflicting extent is one in which the write operation attempts to update an extent.

To improve the probability of a read hit, the controller can *prestage* data into its cache. Prestaging means that data is read into the controller's cache ahead of its actually being required for use by an application. The amount of data that is prestaged depends on (1) whether the data is being accessed in a direct (random) mode or in a sequential mode and (2) the controller model and the enhancements made to the controller.

C For *direct mode*, after the record is located, the 3390-3 and 3990-6 (initial version) stages in the balance of the track being read.

The 3990 Model 6 (with record cache) stages only the records requested into cache, eliminating the balance of the track staging that is normal with track caching as was implemented on initial versions of 3990-6 and on the 3990-3. This improvement reduces the PEND time caused by the controller busy during track staging.

C As examples of prestaging for *sequential mode*, the 3990-3 reads up to two tracks into the cache[5] before they are required, while the ESS 2105 sequential staging reads up to two cylinders ahead.

During prestaging operations for sequential reads, the control unit regularly checks to see whether other I/O requests are waiting to be processed. If any are waiting, the control unit interrupts the prestage operation, processes the queued requests, and continues with the prestage.

In DASD Fast Write Mode, the data is stored simultaneously in cache storage and in nonvolatile storage (NVS). At some subsequent time, the data in NVS can be *destaged* to DASD.

In Cache Fast Write Mode, data is placed into cache immediately, and there is no interaction with the device nor with NVS. However, if cache memory is required (or if Cache Fast Write Mode is turned off), the data in cache is destaged to DASD. Significant PEND time can result from destaging to DASD.

- **Dual Copy Initialize**. Large PEND times for IBM-3390 devices may be caused by dual copy initialize. IBM recommends the following[6] for best system performance when using Dual Copy Initialize:

"C Use enhanced dual copy, that is, set DASD fast write on to all of your dual copy pairs.

[5] With the Sequential Staging Performance Enhancement, the 3990-3 can prestage up to a full cylinder (15 tracks) into the cache.

[6] Source: IBM 3993/9390 Introduction.

C The write hit ratio should be 90% or higher for good performance. Write hit ratios are normally 99-100% if microcode supporting Record Cache II has been installed.

C The read to write ratio should be 2:1 or greater for dual copy candidate volumes.

C Wherever possible, use DLSE operations for your pairs.

C As much as is possible, spread your dual copy pairs across multiple storage controls. By doing so, you lessen the impact of having a larger number of fast dual copy pairs on one subsystem, especially for a heavily loaded DASD subsystem. Remember that both devices in a duplex pair must be in the same logical DASD subsystem.”

Chapter 1.3: DISC time

DISC means that there is some delay that is often (but not always) associated with a mechanical movement during which the device disconnects from the control unit.

With legacy systems (e.g., 3380 drives attached to 3990-2 control units), the DISC time of most concern was associated with seek (arm movement) and rotational position sensing (time waiting for the disk platter to rotate to the location where desired data resides). Considerable performance improvement efforts were directed at reducing the seek activity and reducing the rotational position sensing (RPS)[7] delays for the legacy systems. These two mechanical delays still exist for most modern *redundant array of independent disks* (RAID)[8] systems, but their impact can not be directly reduced with normal methods.

With modern disks, data is cached into Actuator Level Buffers (ALBs), that contain data read from a track on the disk platter. Using ALBs can eliminate the RPS delays for records read on a particular track, since required data is read into the device buffer during a single rotation and stored until a path is available to transfer the data. However, if a record is to be read from a new track, some RPS delay could exist since the record would not be in the ALB, and must be read from the new track. Some initial RPS delay would apply in this case. This initial RPS delay is neither measured nor preventable.

Additionally, data is cached into increasingly large cache on the controller. For a read operation, desired data often is found in the cache. Write operations normally end as the date to be written is placed in non-volatile storage (NVS); and the storage processor writes

[7] RPS delays are caused by a path not being available when the required data came under a device read head. Since a path was not available, the data could not be read and another rotation of the platter was experienced until the data again came under the device read head. Multiple rotations might be required, depending on the busy level of the path.

[8] An array is an ordered collection of physical devices (disk drive modules) that are used to define logical volumes or devices.

the data to the device asynchronous with other activity (as a “back end” staging operation). The write activity can result in DISC time.

Consequently, DISC time for modern systems is a result of *cache read miss* operations, potentially back-end staging delay for write operations, peer-to-peer remote copy (PPRC) operations, and other miscellaneous reasons[9]. DISC time often can be very small with adequate cache. For example, there would be zero disconnect time for a cache read hit (the record was found in the cache).

Chapter 1.4: CONN time

CONN time includes the data transfer time, but also includes protocol exchange[10] (or "hand shaking") between the various components at several stages of the I/O operation.

For devices attached to paths that include parallel channels and ECON channels, the data transfer time is simply the number of bytes transferred divided by the transfer speed. This is because a parallel channel or ESCON channel can have only one data transfer operation in execution at one time.

For devices attached to paths that include FICON channels, the algorithm is more complicated. This primarily is because a FICON channel can perform multiple data transfer (read and write) operations at one time. The data packets for multiple read or write operations are interleaved (or multiplexed) in the FICON link. CONN time for an individual I/O begins with the first frame of data transferred and ends last frame of data transfer, even though data for other I/O operations might be transferred concurrently on the link. Consequently, if multiple data packets (representing data for multiple read or write operations) are interleaved on the FICON link, the elapsed time for any particular I/O operation can be elongated[11] when compared with the elapsed time of the same I/O operation on an ESCON channel.

Chapter 1.5: OTHER time

There are at least two other potential I/O delays for DASD: (1) waiting for the I/O completion interrupt to be serviced by a processor and (2) waiting for the I/O interrupt to

[9] Artis has described a “sibling PEND” condition that results from collisions within the physical disk subsystem of RAID devices. See “Sibling PEND: Like a Wheel within a Wheel,” www.cmg.org/cmgpap/int449.pdf.

[10] Note that the protocol exchange occurs at multiple points in the normal I/O operation, even though it is shown only once in this exhibit.

[11] The relative speed of a FICON channel is much higher than that of an ESCON channel. Consequently, the elapsed time of any particular I/O operation should be less on a FICON channel than on an ESCON channel, even if there are multiple I/O operations interleaving data. This statement regarding elapsed time is not necessarily correct if a large number (more than 5) I/O operations are concurrently executing on a FICON channel. Dr. H. Pat Artis and Mr. Robert Ross have presented the results of research indicating that performance degrades significantly when more than 5 I/O operations (Open Exchanges) are concurrently active on a FICON channel (see “Understanding FICON Channel Path Metrics”at www.perfassoc.com).

be serviced by a domain under PR/SM. Neither potential I/O delay is expected to be of the magnitude of the four "standard" I/O delays. However, they can be significant in special circumstances.

- Multi-processor configurations running under MVS can use any processor to service an I/O interrupt. However, when a processor services an I/O interrupt, the processor's high-speed cache storage is no longer valid when control is returned to the interrupted task. Consequently, many of the processor's high-performance design features may be nullified.

  A hardware feature allows processors to be disabled for I/O interrupts. With this method, only a small number (perhaps only one) processor is enabled for interrupt processing. Only this processor will have its high-speed cache storage disturbed by the task-switching required for interrupt processing, and only this processor will periodically have its high-performance design features nullified. The disadvantage to this approach is that an interrupt may occur while the processor is busy servicing a previous interrupt.

  If an interrupt is pending and no processor is enabled to service the interrupt, the interrupt must wait until a processor is available. This time should be insignificant, unless the system is processing a significantly large number of I/O operations. If the system is processing a large number of I/O operations, the interrupt pending delay could pose performance problems.

  After the processor completes processing for an I/O interrupt, it issues a Test Pending Interrupt (TPI) instruction to determine whether there are any interrupts pending. If an I/O interrupt is pending, the processor proceeds to service that interrupt.

  The IEAOPTxx member of SYS1.PARMLIB contains the **CPENABLE** keyword. This keyword specifies the percent of I/O interrupts detected by the TPI instruction, compared with all I/O interrupts. When the percent exceeds the high threshold of the CPENABLE keyword, MVS enables another processor to handle pending I/O interrupts. If the percent falls below the low threshold of the CPENABLE keyword, MVS will disable a processor (to the point that only one processor is enabled). The low and high threshold values for CPENABLE are 10 and 30 percent, respectively. These values normally mean that less than 30% of the I/O interrupts will be delayed for I/O interrupt service.

- MVS environments running under as a guest under VM or in a logical partition (LPAR) under PR/SM are subject to I/O interrupt delays. These delays can occur if another guest (for VM) or another domain is in its dispatch interval when the I/O interrupt completion is posted. The I/O interrupt remains pending until the guest or domain is dispatched. These delays have been estimated to be far more significant than might otherwise be expected.

Neither of the potential I/O delays described above is measured by RMF (although RMF does provide information on the number of I/O interrupts serviced by each processor and the number of TPI instructions resulting in I/O interrupt servicing).

The potential I/O delays are included in this discussion of general DASD performance considerations because (1) they may become important under certain situations and (2) techniques may be developed to assess their impact.

Chapter 2: RMF Data Analysis Considerations

This chapter highlights some of the factors that must be considered when analyzing DASD information collected and recorded by SMF in Type 30 records or by RMF in Type 70(series) records.

These factors do **not** preclude a comprehensive analysis of performance data and usually do not prevent insight into the causes of unacceptable performance. However, the factors must be recognized and accounted for both by CPExpert in analyzing data and by the user in reviewing CPExpert's results. The factors stem from (1) the way in which SMF and RMF create and record Type 30 and Type 70(series) information, and (2) inherent limitations caused by data averages.

Chapter 2.1: SMF information

SMF Type 30 records contain a record sub-type code to identify when the records are written:

CODE	**SUB-TYPE DESCRIPTION**
1	Job start
2	Interval records
3	Step termination
4	Step total
5	Job termination
6	System address space

SMF will optionally record the different sub-types, depending upon parameters contained in the SMFPRMxx member of SYS1.PARMLIB. Most installations collect Sub-type 4 (Step total) records, and many installations collect Sub-type 2 (Interval) records. If Interval records are recorded by SMF, Sub-type 3 (Step termination) records are automatically created.

It is highly desirable to collect Interval/Step termination information. It is virtually impossible to analyze system performance based upon Step total information if there exists long-running jobs. This is because it is impossible to correlate the information reflected in the Step total records with the information contained in SMF Type 70(series) data.

The Sub-type 4 records are written only after a long-running job step terminates, while the SMF Type 70(series) records are written at user-defined intervals (the interval typically is every 30 minutes or so). Long-running job steps may span many RMF recording intervals [RMF is responsible for creating the SMF Type 70(series) records]. Consequently, there may be many RMF interval records written between the start and end of a long-running job step.

Sub-type 2 (Interval) records are written at user-defined intervals (typically the interval selected is the same interval as the RMF interval records). SMF writes a Sub-type 2 record when the specified interval has lapsed after the start of the job step and continues to write Sub-type 2 records at each subsequent interval. When the job step terminates, SMF writes a Sub-type 3 record containing the information since the last Sub-type 2 record was written. One consequence of the interval records is that system usage can be identified by workload, and can be correlated with the overall system statistics recorded by RMF in the SMF Type 70(series) records.

There are two variations in how SMF and RMF write interval data: (1) non-synchronized and (2) synchronized. Synchronization of SMF and RMF records is an option that must be explicitly specified in the SMFPRMxx member of SYS1.PARMLIB.

C **Non-synchronized writing of SMF and RMF interval data.** With non-synchronized writing of SMF and RMF interval data, the Sub-type 2 records are written based upon the interval lapse from the start of the job step. They are not written at the same time as is the SMF Type 70(Series) records. This lack of coordination between recording the two record types poses a correlation problem: a particular Sub-type 2 (or Sub-type 3) record may span between two RMF recording intervals. From a data analysis view, there is no way to precisely determine whether the data reflected in the Sub-type 2 record (or Sub-type 3 record) should belong to the first RMF Type 70(series) interval or should belong to the second RMF Type 70(series) interval.

For example, suppose that the RMF recording interval were specified as 30 minutes, and RMF was directed to synchronize on the hour and half-hour. The RMF data would be collected and recorded at 10:00, 10:30, 11:00, 11:30, etc. Further suppose that a particular job step started at 15 minutes past the hour. Assuming that the Type 30 interval recording were specified as 30 minutes, SMF would create a Type 30 (Sub-type 2) interval record at 45 minutes past the hour, 15 past the next hour, and so forth. Thus, the RMF data would be recorded on the hour and half-hour, while the Sub-type 2 data would be recorded "offset" by 15 minutes.

CPExpert addresses this problem by pro-rating the SMF Type 30 information based upon elapsed time. In the above example, 50% of the actual workload data contained in the SMF Type 30 (Sub-type 2 or Sub-type 3) records would be attributed to one RMF measurement interval and 50% would be attributed to the next RMF measurement interval. This pro-rating approach essentially assumes that the resources required by a job step do not vary much from one instant to the next.

This approach works quite nicely so long as the job step uses resources in a uniform fashion. Many job steps exhibit this characteristic, and the resources required by the job step do not vary much as the job executes. Resources distributed using the pro-rating approach result in fairly consistent usage characteristics when comparing the summarized Type 30 data with SMF Type 70(series) data.

However, some job steps exhibit significant cycles, or require resources at the beginning or end of the job step. For these job steps, the pro-rating approach does not properly distribute the resource usage into the correct RMF measurement interval. Summarized Type 30 data would not compare well with Type 70(series) data if many job steps exhibit this cyclic or burst nature of resource usage. Unfortunately, there is no way to better distribute the data. Consequently, analysis based upon Type 30 data must be viewed with some caution. The analysis **generally** will be sufficiently precise for performance analysis purposes. However, anomalies will appear and results must always be subjected to a "reality" test.

This point is significant for the DASD Component, because the DASD Component attributes DASD usage to workloads based upon correlating SMF Type 30 data with SMF Type 74 data. The DASD I/O activity at the job step level is obtained from the SMF Type 30 interval records (using a modification to MXG or to MICS). This DASD I/O activity is pro-rated to the RMF measurement intervals as described above. The RMF DASD device I/O characteristics (IOSQ, PEND, DISC, and CONN times) are attributed to workloads based upon the pro-rating.

It is possible that the pro-rating method will result in improper attribution of I/O device characteristics to workloads. For example, suppose that a job step completed a few minutes past the hour (and that RMF data records were synchronized on the hour and half-hour). When the job step completed, it could execute many DASD I/O operations. These I/O operations would mostly be attributed to the previous RMF interval and the DASD device characteristics of that interval would be associated with most of the I/O operations. Suppose that there were no I/O problems with the device in the first interval. It is possible, however, that as the job step completed, it experienced significant DASD I/O problems that would be reflected in the second RMF interval. Since only a few of its I/O operations would be attributed to the second RMF interval, CPExpert would associate the I/O problems to only a few of its I/O operations. Consequently, CPExpert might consider the job step (and the workload category associated with the job step) to receive good DASD service, when the workload actually received bad service because of its burst I/O operations.

This example is not intended to invalidate the techniques CPExpert uses. Rather, the example is presented to explain a unique situation in which the techniques could result in improper conclusions. Readers may note that IBM's Service Level Reporter and any other software analyzing SMF/RMF data face the same analysis problem. The problem is with the data; not with the technique.

C **Synchronized writing of SMF and RMF interval data.** Synchronized writing of SMF and RMF was introduced with MVS/ESA SP4. When interval accounting is synchronized, SMF generates interval records for a work unit based on the end of the

SMF global recording interval, rather than the start time of a job. This feature allows Type 30 records (and other record types) to be synchronized with writing RMF Type70(series) records. SMF places indicators (or “flags”) in SMF Type70(series) records to indicate whether SMF and RMF records are synchronized.

It is not necessary for CPExpert to pro-rate data if the recording intervals are synchronized.

Chapter 2.2: Data Averages

The data collected by RMF and recorded in SMF Type 70(series) records provide a valuable source of information about the use and interaction of system components and workloads. However, the data are summarized and recorded at specific intervals (e.g., every 30 minutes). For most data elements, analysis must be accomplished based upon the **summary** or **average** values.

For example, the DASD IOSQ time reported for each device is the total for the measurement interval. The average IOSQ time per I/O operation is computed by dividing the total IOSQ time by the number of I/O operations. This average may have no relation to the IOSQ time experienced by any particular I/O operation. This problem is particularly pervasive as the RMF recording interval becomes more lengthy (e.g., if the recording interval were 60 minutes). The DASD IOSQ time may be quite long during the first half of the RMF measurement interval when contending workloads execute. The DASD IOSQ time may be short during the last half when a workload executes without contention from other applications. The average of the two extremes may lead to a conclusion that there was no problem with DASD IOSQ time for the entire interval!

Most DASD analysis performed by CPExpert assumes either a uniform or an exponential distribution of DASD I/O operations. For example, the pro-rating discussed in the previous chapter assumes a uniform distribution of I/O operations on a **job step** basis, over the life of the job step. However, the queuing models employed by CPExpert to analyze various aspects of DASD delays generally assume an exponential distribution of I/O operations at the **device** level, over an entire RMF measurement interval. Neither of these assumptions may be correct.

Wicks[12] illustrates a variety of distributions of the arrival rate of events, ranging from uniform distribution, to "cafeteria" distribution (events mostly arrive in clusters), to "London bus" distribution (events arrive only in clusters), to a random distribution (events exhibit a Poison or exponential arrival). The arrival of many events in computer systems exhibit an exponential distribution, and M/M/1 or M/M/C queuing models can fairly represent many aspects of the systems.

[12] Wicks, R. J., "*Balanced Systems and Capacity Planning*, IBM Corporation Washington Systems Center Technical Bulletin GG22-9299-02

However, Wicks gives an excellent example of exceptions: when editing a dataset using ISPF, the entire dataset may be read, some time is spent editing, and then the entire dataset may be written. The I/O requests in this instance would be similar to Wicks' "London Bus" distribution.

There is a tradeoff between (1) recording RMF data frequently, incurring the overhead and storage requirements of the additional RMF records, and requiring additional resources to analyze the data versus (2) recording RMF data less frequently, having less precise or representative data to analyze, and minimizing the resources required to perform the analysis. The importance of these tradeoffs must be evaluated in light of the objectives of the analysis and the requirements for precision of results.

In any event, any review of analysis and conclusions (whether by CPExpert or by a performance analyst) must be viewed with some caution because of the data summary, data averaging, and data distribution issues.

If the analysis consistently results in the same conclusions, you can be reasonably sure that the analysis is correct. However, it generally is unwise to make changes based upon analysis of a single day's RMF measurement information unless a "reality test" indicates that the analysis clearly is correct.